

Remarks

Claims 1, 3-18, and 20-27 are pending in the present application. Claims 2, 19, and 28-39 are canceled, as indicated in the above listing of the claims. The independent claims are 1, 11, 12, and 18.

Claims 1, 3-8, 10-15, 17, 28-35, 37, 38, and 39 stand rejected under 35 USC 103(a) as being unpatentable over Rangachar et al. (US 6,301,252) in view of Barret et al. (US 6,782,420).

Regarding claim 1, Examiner states that Rangachar et al. disclose a method of managing a telecommunications network device, comprising:

c) receiving the command at the command interface from a user interface; (see Rangachar col. 4, lines 28-30: command is sent to network manager which is central command processing software)

d) forwarding the command to the application; (see Rangachar col. 5, lines 23-28): send to process (command proxy) for applicable network device); and

e) completing execution of the command; (see Rangachar col. 4, lines 41-47: process command at network device).

Examiner also states that Rangachar et al. disclose a network management system for controlling a plurality of managed network devices, (see Rangachar col. 4, lines 7-11: "... a network manager (sometimes referred to as a network server or a network management station) communicating with a plurality of ... switches connected by communications links ..."). Rangachar et al. do not specifically disclose the storage (register) of a command within a command interface. However, Barrett et al. disclose:

a) registering at least one command executable by an application with one of a plurality of distributed command proxies associated with a command interface (see Rangachar col. 7, lines 6-10; col. 7, lines 14-18: plurality of processes (i.e. daemons or executions) for distributed command interfaces), said command proxy being local to the application; (see Barrett col. 2, lines 21-27; col. 6, lines 6-11; col. 6, lines 53-55; col. 1, lines 45-47: distributed environment, store (register) commands within the command interface); and

b) registering the command through the command proxy local to the application with a central command daemon associated with said command interface (see Barrett col. 2, lines 21 - 27; col. 6, lines 6-11; col. 6, lines 53-55; col. 1, lines 45-47: distributed environment, store (register) commands within the command interface).

According to Examiner, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Rangachar et al. to enable the registration of a command as taught by Barrett et al.

Applicants were unable to find Examiners cited reference (Rangachar col. 7, lines 6-10; col. 7, lines 14-18) that had any correlation with claim 1 of the present application: a) registering at least one command executable by an application with one of a plurality of distributed command proxies associated with a command interface, said command proxy being local to the application. Perhaps Examiner was referring to Barrett et al. col. 7, lines 6-10, which states "When told to process the command, the command object does two things: it registers acknowledgment and response filters with the event distributor 46 on behalf of the client, and it passes information through the MO to the EMS MEDIATOR 60. The MEDIATOR is a SNMP MEDIATOR in this scenario, so that information can be sent off to the appropriate network element for eventual execution." Examiner also states that Barrett et al. col. 2, lines 21 -27; col. 6, lines 6-11; col. 6, lines 53-55; col. 1, lines 45-47 teaches: b) registering at least one command executable by an application with one of a plurality of distributed command proxies associated with a command interface, said command proxy being local to the application. The reference teaches at col. 2, lines 21 -27 "In keeping with another important aspect of the invention, complex commands are supported. The element server receives and stores a chain of commands including at least one contingent command, determines the results of at least one of the commands, based upon determined results selectively taking action in accordance with the results of the at least one of the commands"; col. 6, lines 6-11 state "When a command object 52 is created, it requests to be placed in queue by the command handler 40. This is how the command handler is able to keep track of all the commands within the system." Barrett et al. do not teach the use of "a plurality of distributed command proxies", but the fact that a command

object 52 is created, it requests to be placed in queue by the command handler. Barrett's queue is substantially the antithesis of Applicants' distribution. Neither Barrett et al. nor Rangachar et al. teach a plurality of distributed command proxies associated with a command interface, said command proxy being local to the application. Rangachar et al. teach a plurality of managed *network devices*, not a plurality of distributed *command proxies*. One is hardware, one is software. Claim 1 is prima facie not obvious in light of the combination of references, as the references teach a centralized system and Applicants teach a distributed localized system.

Regarding claim 3, Examiner states that Rangachar et al. disclose the method of claim 1, wherein receiving the command at the command interface from a user interface and forwarding the command to the application comprises:

- a) receiving the command at one of the plurality of command proxies that is local to the user interface; (see Rangachar col. 4, lines 28-30; col. 5, lines 23-28: commands are diverted to central network manager (central command daemon), then command is sent to specific process (command proxy) for the particular network device);
- b) determining if the application that registered the received command is local to the command proxy that is local to the user interface; (see Rangachar col. 5, lines 23-28; col. 4, lines 41-47: command is sent to specific process (command proxy) for the particular network device from network manager (central command daemon)); and
- c) if yes, then forwarding the received command to the application that registered the received command; and if no, then forwarding the received command to the central command daemon; (see Rangachar col. 4, lines 28-30: forward commands to network manager for command implementation at local process, col. 5, lines 23-28: forward to network device for processing).

Applicants point out that in Rangachar col. 4, lines 28-30 a request is directed to the network manager, and that with Applicants invention the requests are received locally, and if they can be handled locally then they are, otherwise they are sent to the central command daemon. Rangachar et al. are teaching a centralized system, while Applicants are teaching a distributed localized interface.

Regarding claim 4, Examiner states that Rangachar et al. disclose the method of claim 3, further comprising:

a) forwarding the received command to the one of the plurality of command proxies that registered the received command; (see Rangachar col. 5, lines 23-28: forward to process (command proxy)); and

b) forwarding the received command to the application that registered the received command; (see Rangachar col. 4, lines 41-47: forward to network device for processing).

The pattern is consistent, Applicants, unlike Rangachar et al., teach distributed localized control, while Rangachar et al. (at col. 5, lines 23-28 and col. 4, lines 41-47) teach a centralized system. The citations bear this out. “The network manager may then determine necessary control information needed to carry out a network operation by creating a plurality of processes, one process for each of the switches, and exchanging information among the processes partially based on the state model” and “The network manager processes the request according to control and management procedures at the network manager to identify generic commands to transmit to one or more switches in the network. The generic commands are interpreted at each of the one or more switches to produce a respective instruction to direct operation of each of the one or more switches.” Rangachar’s network manager controls all, while Applicants’ is handled through the plurality of command proxies.

Regarding claim 5, Examiner states that Rangachar et al. does not specifically disclose the storage (register) of a command within a command interface. However, Barrett et al. disclose the method of claim 1, wherein the command interface is a central system and wherein registering at least one command executable by an application with a command interface comprises: registering the command with a central command daemon; ((see Barrett col. 2, lines 21-27; col. 6, lines 6-11; col. 6, lines 53-55; col. 1, lines 45-47: distributed environment, store (register) commands within the command interface) see Barrett col. 3, lines 16-20).

Applicants have previously discussed that Barrett et al. teach “This invention provides an application programming interface (API) and protocol that provides for efficient

communication between a distributed client application and an element management server independent of the communication protocol to the network element. An Element Management Application Programming Interface (EMAPI) provides benefits over known management systems as noted above.” There is no mention of a local command proxy, nor of registering a command through the command proxy local to the application. Claim 5 is a dependent claim depending from claim 1, where claim 5 merely adds that there can be a connection to a command interface that is a central system.

Regarding claim 6, Examiner states that Rangachar et al. do not specifically disclose an API interface for application development. However, Barrett et al. disclose the method of claim 1, wherein completing execution of the command comprises:

a) receiving the command through a command application programming interface (API) linked into the application; and calling a call back routine within the application corresponding to the received command; (see Barrett col. 2, lines 41-46; col. 3, lines 16-20; col. 4, line 64 - col. 5, line 2: API interface exists for server network management system for application development).

Regarding claim 7, Examiner states that Rangachar et al. do not disclose a display interface for responses to command. However, Barrett et al. disclose the method of claim 6, wherein completing execution of the command further comprises: calling a display routine linked into the application to send any display data directly to the user interface; (see Barrett col. 2, lines 38-41; col. 5, lines 56-60: display commands and responses at command console).

Regarding claims 8, 15, and 35, Examiner states that Rangachar et al. do not specifically disclose a web interface for the network management system. However, Barrett et al. disclose the method of claims 1, 13, and 32, wherein the user interface comprises: a web interface; (see Barrett col. 3, lines 49-52; col. 4, lines 16-22: web type command interface for network management system).

Applicants’ claims 6, 7, and 8 are all dependent claims depending on claim 1 or an intervening dependent claim. Claim 15 is a dependent claim depending on independent claim 12 via dependent claim 13. Claim 35 is a dependent claim depending on independent claim 30 via

dependent claim 32. Claims 15 and 35 are addressed appropriately when the Examiner addresses the parent claims.

Applicants' claims 6 recites: receiving the command through a command application programming interface (API) linked into the application; and calling a call back routine within the application corresponding to the received command. Barrett et al. teach that the management server generates a web page. Applicants are not generating a web page, but calling a call back routine. The cited reference is not relevant.

Applicants' claims 7 and 8 recite: calling a display routine linked into the application to send any display data directly to the user interface, and a web interface. Applicants' application is local, while Barrett's teaches the use of a centralized management server to generate a web page.

Regarding Claims 10, 17, and 37, Examiner states that Rangachar et al. disclose the method of claims 1, 13 and 32, wherein the user interface comprises: a network/element management system interface; (see Rangachar col. 4, lines 7-11: network management system with management console or station).

Applicants' claim 10 is a dependent claim depending from claim 1, while claims 7 and 37 depend from independent claims 12 and 30. Claim 10 recites a network/element management system interface. Claim 10 has all the limitations of claim 1, and stands on the basis of the previously discussed elements of claim 1.

Regarding claim 11, Examiner states that Rangachar et al. disclose a method of managing a telecommunications network device, comprising:

c) receiving the command at a user interface; (see Rangachar col. 4, lines 28-30: forward command to network manager (central command processing));

d) forwarding the command to a second command proxy, wherein the second command proxy is local to the user interface; (see Rangachar col. 5, lines 23-28: forward command to applicable process (command proxy));

e) forwarding the command through the second command proxy to the central command daemon; (see Rangachar col. 4, lines 28-30: forward command to network manager (central command processing));

f) forwarding the command through the central command daemon to the first command proxy; (see Rangachar col. 5, lines 23-28: forward command to applicable process (command proxy));

g) forwarding the command through the first command proxy to the application; (see Rangachar col. 4, lines 28-30: forward command to network manager (central command processing)); and

h) completing execution of the command, (see Rangachar col. 4, lines 41-47: process command at network device).

Examiner also states that Rangachar et al. do not specifically disclose the storage (register) of a command within a command interface. However, Barrett et al. disclose a method of managing a telecommunications network device, comprising:

a) registering at least one command executable by an application with a first command proxy, wherein the first command proxy is local to the application; (see Barrett col. 2, lines 21 - 27; col. 6, lines 6-11; col. 6, lines 53-55; col. 1, lines 45- 47: distributed environment, store (register) commands within the command interface); and

b) registering the command through the first command proxy with a central command daemon; (see Barrett col. 2, lines 21-27; col. 6, lines 6-11; col. 6, lines 53-55; col. 1, lines 45-47: distributed environment, store (register) commands within the command interface).

Examiner asserts that it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Rangachar et al. to enable the registration of a command as taught by Barrett et al. One of ordinary skill in the art would be motivated to employ Barrett et al. in order to enable efficient communications between a management server and distributed client, (see Barrett col. 3, lines 16-20).

In contrast to Barrett et al., Applicants' claim 11 recites registering at least one command executable by an application with a first command proxy, wherein the first

command proxy is *local to the application*. Barrett et al. do not have a counterpart. Barrett et al. (col. 2, lines 21 –27) teach “In keeping with another important aspect of the invention, complex commands are supported. The element server receives and stores a chain of commands including at least one contingent command, determines the results of at least one of the commands, based upon determined results selectively taking action in accordance with the results of the at least one of the commands” (col. 6, lines 6-11; col. 6, lines 53-55; col. 1, lines 45-47). The element server (EMS) 24 is a centralized server, as shown in Fig. 1.

Barrett et al. (col. 6, lines 6-11) teach that when a command is created, it requests to be placed in the queue. Applicants do not have a “queue”. Barrett et al. (col. 1, lines 45-47) teach “In accordance with the invention this is achieved by supporting command execution in a distributed environment. With respect to one aspect of the invention commands can be composed in a different process or processor by utilizing CORBA or CORBA-like communication. Commands are independent of the protocol used between the managing computer and the element being controlled. Preferably, the commands are composed in EMAPI notation and the target protocol is SNMP.” SNMP is an acronym for Simple Management Network Protocol, CORBA is an acronym for Common Object Request Broker Architecture, and EMAPI is an acronym for Element Management Application Programming Interface. In contrast, Applicants do not claim that the commands are independent of the protocol used between the managing computer and the element being controlled, but that the commands are sent to multiple proxies, as well as the central daemon. Instead of a Common Object Request Broker Architecture, Applicants are claiming a decentralized localized distribution system. Therefore, the rejection is respectfully overcome.

Regarding claim 12, Examiner states that Rangachar et al. disclose a method of managing a telecommunications network including a first network device and a second network device, comprising:

a) executing a community command daemon on one of the first or second network devices; (see Rangachar col. 4, lines 28-30: multiple network devices are managed by network manager (central command daemon)); and

b) executing a first application on the first network device; executing a second application on the second network device; (see Rangachar col. 5, lines 23-28: applicable process (command proxy) is executed),

Examiner also states that Rangachar et al. do not specifically disclose the storage (register) of a command within a command interface. However, Barrett et al. disclose a method of managing a telecommunications network including a first network device and a second network device, comprising:

c) registering a first command executable by the first application with a first command interface on the first network device; registering a second command executable by the second application with a second command interface on the second network device; (see Barrett col. 2, lines 21-27; col. 6, lines 6-11; col. 6, lines 53-55; col.1, lines 45-47: distributed environment, store (register) commands within the command interface) and

d) registering the first and second commands with the community command daemon, (see Barrett col. 2, lines 21-27; col. 6, lines 6-11; col. 6, lines 53-55; col.1, lines 45-47: distributed environment, store (register) commands within the command interface).

According to Examiner, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Rangachar et al. to enable registration of a command as taught by Barrett et al. One of ordinary skill in the art would be motivated to employ Barrett et al. in order to enable efficient communications between a management server and distributed client; (see Barrett col. 3, lines 16-20).

Applicants' claim 12 recites an element constituting a decentralized registering of commands and applications. Barrett et al. do not have a counterpart (col. 2, lines 21 -27). Barret et al. teach "In keeping with another important aspect of the invention, complex commands are supported. The element server receives and stores a chain of commands including at least one contingent command, determines the results of at least one of the

commands, based upon determined results selectively taking action in accordance with the results of the at least one of the commands”; col. 6, lines 6-11; col. 6, lines 53-55; col. 1, lines 45- 47). The element server (EMS) 24 is a centralized server, as shown in Fig. 1.

Barrett et al. (col. 6, lines 6-11) teach that when a command is created, it requests to be placed in the queue. Applicants do not have a “queue”. Barrett et al. (col. 1, lines 45-47) teach “In accordance with the invention this is achieved by supporting command execution in a distributed environment. With respect to one aspect of the invention commands can be composed in a different process or processor by utilizing CORBA or CORBA-like communication. Commands are independent of the protocol used between the managing computer and the element being controlled. Preferably, the commands are composed in EAPI notation and the target protocol is SNMP.” SNMP is an acronym for Simple Management Network Protocol, CORBA is an acronym for Common Object Request Broker Architecture, and EAPI is an acronym for Element Management Application Programming Interface. In contrast, Applicants do not claim that the commands are independent of the protocol used between the managing computer and the element being controlled, but that the commands are sent to multiple proxies, as well as the central daemon. Instead of a Common Object Request Broker Architecture, Applicants are claiming a decentralized localized system of applications and commands. Therefore, the rejection is respectfully overcome.

Regarding Claims 13 and 14, Examiner states that Rangachar et al. disclose the method of claim 12, further comprising:

- a) receiving the first/second command at the community command daemon from a user interface; (see Rangachar col. 6, lines 24-26: multiple command interface systems ;col. 4, lines 28-30: network manager (community command daemon));
- b) forwarding the first/second command through the community command daemon to the first/second command interface; (see Rangachar col. 5, lines 14-19: first/second application);
- c) forwarding the first/second command through the first/second command interface to the first/second command interface(see Rangachar col. 5, lines 23-28); and

d) completing execution of the first/second command; (see Rangachar col. 4, lines 41-47: process command at network device).

Applicants' claims 13, 14, 15 and 17 are all dependent claims depending on claim 12 or an intervening dependent claim. Claim 15 is a dependent claim depending on independent claim 12 via dependent claim 13, and claim 17 is a dependent claim depending on independent claim 12 via dependent claim 13. Claims 13, 14, 15, and 17 have all the limitations of claim 12, and they stand on the basis of the previously discussed elements of claim 12.

Claims 28-35 and 37-39 stand rejected under 35 USC 103(a) as being unpatentable over Rangachar et al. (US 6,301,252) in view of Barrett et al. (US 6,782,420).

Claims 28-35 and 37-39 are canceled.

Claims 9, 16, 18, 20-27, and 36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Rangachar-Barrett as applied to claims 1, 12, 18, and 30 above, and further in view of Chen et al. (US 6,625,590).

Regarding claims 9, 16, 26, and 36 Examiner states that Rangachar et al. do not specifically disclose a command line interface. However, Chen et al. disclose the method of claims 1, 13, 18, and 32, wherein the user interface comprises: a command language interface (CLI); (see Chen col. 1, lines 50-56: command line interface for command input).

According to Examiner, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Rangachar et al. to utilize a command line interface for usage by the network management system as taught by Chen et al. One of ordinary skill in the art would be motivated to enhance Rangachar et al. in order to enable a flexible and robust user interface for network management control.

Applicants have canceled Claim 36. Claim 9 is a dependent claim depending from claim 1, and has all the limitations and novelty of claim 1. Applicants respectfully submit that, in light

of the arguments, the rejection is traversed. Claim 16 is a dependent claim depending from dependent claim 13, which is depending from claim 12, and has all the limitations and novelty of claims 12 and 13. Applicants respectfully submit that, in light of the arguments for claim 12 and 13, the rejection is traversed. Claim 26 is a dependent claim depending from claim 18, and has all the limitations and novelty of claim 18. Applicants respectfully submit that, in light of the arguments, the rejection is traversed. None of the independent claims are cited Chen et al.

Regarding claim 18, Examiner states that Rangachar et al. disclose a telecommunications network device, comprising:

a) an application capable of executing a command; (see Rangachar col. 5, lines 23-28: command processes (application) to execute a command).

Examiner also states that Rangachar et al. do not specifically disclose a common command interface. However, Chen et al. disclose:

b) a common command interface comprising a distributed system having a central command daemon and a plurality of distributed command proxies; (see Rangachar col. 4, lines 18-27; col. 7, lines 6-10: wherein the application is capable of registering the command with the common command interface and the common command interface is capable of receiving the command from a user interface and forwarding the received command to the application; (see Chen col. 7, lines 55-59: command line interface))

According to Examiner, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Rangachar et al. to enable the capability of a common command interface as taught by Chen et al. One of ordinary skill in the art would be motivated to enhance with Chen et al. in order to enable a flexible and robust user interface for network management control.

Applicants have reviewed the cited reference (Chen col. 7, lines 55-59), and the Examiner is in error. Neither Chen et al. nor Rangachar et al. teach a plurality of distributed command proxies. Chen et al. teach "The network devices coupled to network 68, such as network device 64, can comprise bridges, routers, computers, storage systems, etc. Network

environment 60 further includes file server 66, which in addition to storing network programs and data, can also store UCI command files that may be remotely invoked to perform network management operations.” The only remote operations are “network management operations”. Furthermore, Chen et al. do not teach “wherein the application is capable of registering the command with the common command interface and the common command interface is capable of receiving the command from a user interface and forwarding the received command to the application.” In view of the prior discussion of Rangachar et al., Barrett et al., and Chen et al., Applicants submit that the rejection of claim 18 is prima facie not obvious and is traversed.

Regarding claim 20, Examiner states that Rangachar et al. disclose the telecommunications network device of claim 18, wherein the common command interface comprises a distributed system and a central system including:

a central command daemon; (see Rangachar col. 4, lines 18-22; col. 7, lines 6-10: network manager (central command daemon)) and a plurality of distributed command proxies; (see Rangachar col. 4, lines 7-11; col.5, lines 23-28: processes (command proxies)).

Applicants claim 20 is a dependent claim depending from claim 18, and therefore has all the limitations of claim 18. Rangachar-Barrett-Chen do not teach the utility of a plurality of distributed command proxies. Applicants submit that the rejection of Claim 20 is traversed.

Regarding Claim 21, Examiner states Rangachar et al. do not specifically disclose an API interface for application development. However, Barrett et al. disclose the telecommunications network device of claim 18, wherein the application comprises: a command application programming interface (API) for registering the command with the common command interface and for responding to the command forwarded by the common command interface; (see Barrett col. 2, lines 41-46; col. 3, lines 16-20; col. 4, line 64 - col. 5, line 2: API interface for application development).

According to Examiner, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Rangachar et al. to enable usage of an Application

Programming Interface (API) in application development as taught by Barrett et al. One of ordinary skill in the art would be motivated to employ Barrett et al. in order to enable efficient communications between a management server and distributed client; (see Barrett col. 3, lines 16-20).

Applicants' claim 21 is a dependent claim depending from claim 18, and therefore has all the limitations of claim 18. Rangachar-Barrett-Chen do not teach the utility of a plurality of distributed command proxies. Applicants submit that the rejection of Claim 21 is traversed.

Regarding claim 22, Examiner states that Rangachar et al. disclose the telecommunications network device of claim 21, wherein the command API comprises:

b) a command handler for responding to the command forwarded by the common command interface; (see col. 4, lines 18-25: network manager (command handler) processing commands).

Examiner also states that Rangachar et al. do not disclose the registration of a command. However, Barrett et al. disclose:

a) a registration routine for registering the command with the common command interface; (see Barrett col. 2, lines 21-27; col. 6, lines 6-11; col. 6, lines 53-55; col. 1, lines 45-47: distributed environment, store (register) commands within the command interface).

According to Examiner, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Rangachar et al. to enable registration of a command as taught by Barrett et al. One of ordinary skill in the art would be motivated to employ Barrett et al. in order to enable efficient communications between a management server and distributed client; (see Barrett col. 3, lines 16-20).

Applicants' claim 22 is a dependent claim depending from claim 18, and therefore has all the limitations of claim 18. Examiner has equivocated "a command handler" (col. 4, lines 18-25) to a network manager. Rangachar et al. define the function thusly: "The network manager receives commands directed to the switches and processes these commands with control and

management software located at the network manager to determine generic commands to send to one or more of the switches. Each switch of the network contains a vender-specific command interface which interprets the generic commands, translating them into switch specific instructions, to regulate the switch. By this technique, the network manager can carry out an operation on the network through remote control of the switches.” Applicants’ command handler receives its command from the common command interface (switches) not to the common command interface (switches). The two elements are similar in nomenclature, but opposite in orientation. Applicants submit that the rejection to Claim 22 is traversed.

Regarding claim 23, Examiner states that Rangachar et al. do not specifically disclose an API interface for application development. However, Barrett et al. disclose the telecommunications network device of claim 22, wherein the application further comprises: a call back routine, wherein the command handler calls the call back routine when the command handler receives the command forwarded by the common command interface; (see Barrett col. 2, lines 41-46; col. 3, lines 16-20; col. 4, line 64 -col. 5, line 2: API interface for application development).

According to Examiner, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Rangachar et al. to enable usage of an Application Programming Interface (API) in application development as taught by Barrett et al.. One of ordinary skill in the art would be motivated to employ Barrett et al. in order to enable efficient communications between a management server and distributed client; (see Barrett col. 3, lines 16-20).

Applicants’ claim 23 is a dependent claim depending from claim 18, and therefore has all the limitations of claim 18 and intervening claims 22 and 21. Rangachar-Barrett-Chen do not teach the utility of a plurality of distributed command proxies. Applicants submit that the rejection to Claim 23 is traversed.

Regarding claim 24, Examiner states that Rangachar et al. do not disclose a display interface for responses to command. However, Barrett et al. disclose the method of claim 21,

wherein completing execution of the command further comprises: calling a display routine linked into the application to send any display data directly to the user interface; (see Barrett col. 2, lines 38-41; col. 5, lines 56-60: display commands and responses at command console).

According to Examiner, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Rangachar et al. to enable a user interface for command processing as taught by Barrett et al. One of ordinary skill in the art would be motivated to employ Barrett et al. in order to enable efficient communications between a management server and distributed client; (see Barrett col. 3, lines 16-20).

Regarding claim 25, Examiner states that Rangachar et al. do not specifically disclose a web interface for the network management system. However, Barrett et al. disclose the method of claim 18, wherein the user interface comprises: a web interface; (see col. 3, lines 49-52; col. 4, lines 16-22: web type command interface for network management system).

According to Examiner, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Rangachar et al. to enable a web interface as taught by Barrett et al. One of ordinary skill in the art would be motivated to employ Barrett et al. in order to enable efficient communications between a management server and distributed client; (see Barrett col. 3, lines 16-20).

Regarding claim 27, Examiner states that Rangachar et al. disclose the method of claim 18, wherein the user interface comprises: a network/element management system interface; (see Rangachar col. 4, lines 7-11: network management system with management console or station).

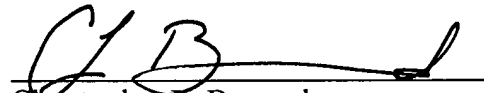
Applicants' claims 25 and 27 are dependent claims depending from claim 18, and therefore have all the limitations of claim 18. Rangachar-Barrett-Chen do not teach the utility of a plurality of distributed command proxies. Applicants submit that the rejections of Claims 25 and 27 are traversed.

Conclusion

Applicants would like to thank Examiner for the attention and consideration accorded the present Application. Should Examiner determine that any further action is necessary to place the Application in condition for allowance, Examiner is encouraged to contact undersigned Counsel at the telephone number, facsimile number, address, or email address provided below. It is not believed that any fees for additional claims, extensions of time, or the like are required. However, if such additional fees are required, Examiner is encouraged to notify undersigned Counsel at Examiner's earliest convenience.

Respectfully submitted,

Date: September 12, 2005



Christopher L. Bernard
Attorney for Applicants
Registration No.: 48,234

F. Rhett Brockington, Ph.D
Agent for Applicants
Registration No.: 29,618

DOUGHERTY, CLEMENTS, HOFER, BERNARD & WALKER

1901 Roxborough Road, Suite 300
Charlotte, North Carolina 28211 USA
Telephone: 704.366.6642
Facsimile: 704.366.9744
cbernard@worldpatents.com